

Non Stochastic Dynamic Programming ____学習の手引き____

池上慧

2015 年 10 月 26 日

概要

この学習の手引きは、DP 学習がスムーズに進むように作成されたものである。理論上重要な証明が載っている文献をネットから拾ってきただけなので偉そうなことは言えないが、注釈に記したページ数に飛ばせば定理等の証明をすぐに参照できるようになっている。一応池上は一通り証明もこなしたので、わからなかったら聞いてください。また、最後の方に、経済学とは直接関係ないが、数理最適化問題の解法としては DP がどのように使われるかがよくわかる例を乗せたのでこちらもご参照ください。iteration が何を行っているのかがよくわかる良い例だと思います。とりあえず京大の経済学のための数学を読めばどうい話なのかわかります。iteration のくだりは縮小写像原理とかあって難しいので、東工大の例でイメージを掴んで、京大の証明をみてください。今回は non stochastic な DP を扱いますが、stochastic 編と iteration の python プログラミングコード編も作ろうと思います。

1 DP 概説

動的計画法 (Dynamic Programming) は、無限期間最適化問題の解法のことである。一般に、無限期間最適化問題が最終的に求めたい物は、無限期間での効用最大化を達成するために取るべき最適行動の列である。ここで DP は、直接に行動の列を算出することで問題を解こうとするのではなく、現在の意思決定を一期前の行動を変数とした関数で表すためにはどのような関数であればよいか、を特定することでこの問題を解こうというものである。目的関数が微分可能であれば、オイラー方程式と横断性条件を使って同様の無限期間最適化問題を解くことができる^{*1}が、そのような仮定を置くことができない場合の一般的な解法としてベルマン方程式を使って解く DP が存在する、と考えると良い。このような事情のため、動的計画法の解説にはオイラー方程式や横断性条件との関係性^{*2}が言及されることも多いが、とりあえず DP を解くということに関してはそれらは関係なく、ベルマン方程式さえ解ければ良い。

ベルマン方程式を解けば DP が解ける^{*3}、ということを理解した次に学ぶべきはベルマン方程式の解き方である。解法として抑えておきたいのは以下の三つである。

- Guess and Verify^{*4}
- Value function iteration^{*5}
- Policy function iteration^{*6}

^{*1} 京大 75 から 82

^{*2} 京大 88、動学最適化 8

^{*3} 京大 82 86、動学最適化 5 8、

^{*4} マクロ 10 から 11

^{*5} マクロ 8 から 9

^{*6} マクロ 11 から 13

上から順に説明していくと、最初の Guess and Verify は value function の形を山勘で当てて、それがベルマン方程式の解となっていることを確認するというものだ。数学的帰納法で漸化式を解く際に山勘で一般項の形を当てはめてそれを正当化する解法が大学受験の際にあったが、あれとやっていることは同じである。次に value function iteration である。これが最も基本的なベルマン方程式の解法である。value function の形を適当に推測し、ベルマン方程式に代入、出力結果を再びベルマン方程式に代入し、といった具合に、入力と出力がほとんど同じになるまでこれを繰り返すことで value function を求めるやり方。これによって value function の形に近づいていけることの証明^{*7}。最後が policy function iteration であり、これは value function と policy function との間で iteration を行うアルゴリズム。policy function が入力と出力であまり変化しなくなるまで続ける。この解法は収束が早いことで知られているが、policy function の真の形に近づくことの証明はなされていない。

2 用語説明

Policy function 日本語では政策関数。DP を解く際に求めたい物であるのは先に述べた通り、一期前の行動と現在の意思決定とをつなげる関数であり、この求めたい関数を policy function と呼ぶ。

Value function 日本語では価値関数。定義としては初期値を変数として目的関数の最大値を表す関数である。すなわち無限期間最適化問題において最大化された目的関数（を初期値の関数として表現したもの）のことである。最適性の原理（後述）により、ある期における状態を入力すると、その状態から達成しうる最大の利得を出力する関数と考えることができる。

Bellman equation 今期の状態を入力すると、今期の行動によって得られる値と今期の行動によって決定される次期の状態を同じ関数に入力して得られる値との和を最大にするような今期の行動を探しだし、その行動をとって得られた最大値を出力するような関数を、解として算出する関数方程式のことである。最適性の原理とは、先の value function がこの bellman equation の解となることを示す定理、すなわち無限期間最適化問題の解をベルマン方程式が与えるということである^{*8}。

3 諸注意と参照もと

この学習の手引きは以下の pdf ファイルを参考にしている。また、各種証明は tex で打つあまりのめんどくささに挫折したので、以下の pdf ファイルから切り取ってきている。

- 経済学のための数学、京都大学経済学研究所、原千秋・梶井厚志
- 動学的最適化入門、工藤教考
- 2013 年度夏学期 上級マクロ経済学講義ノート：動的計画法、阿部修人
- 動的計画法、東工大

特に京大の経済学のための数学は便利であった。

^{*7} 京大 86 から 87

^{*8} 京大 82 から 86

また、このように様々な文献から証明部分を切りはりしているので、問題の定式化が異なっていることが多々ある。注意深く読めば問題ないはずだが、の書き方にふた通りあることには特に注意が必要だ。京大や Stokey and Lucas が採用しているのが利得関数を現在の状態と、次の状態の関数で表されている形態である。こちらは、ベルマン方程式の中に行動という変数（操作変数が出てこない）。二つ目が利得関数を現在の状態と今の意思決定の関数として表したもので、こちらは工藤などが採用している。慣れてくると、この違いは気にならなくなるので、なれるまで読みましょう。